School of Computing and Information Systems
The University of Melbourne
COMP90042
Web Search and Text Analysis (Semester 1, 2019)

Workshop exercises: Week 11

Discussion

1. Using typical dependency types, construct (by hand) a dependency parse for the following sentence: *Yesterday, I shot an elephant in my pyjamas.* Check your work against the output of the online GUI for the Stanford Parser (`http://nlp.stanford.edu:8080/parser/index.jsp`).

2. In what ways is (transition–based, probabilistic) dependency parsing similar to (probabilistic) CYK parsing? In what ways is it different?

3. What is **Discourse Segmentation**? What do the segments consist of, and what are some methods we can use to find them?

4. What is an **anaphor**?

   (a) What is **anaphora resolution** and why is it difficult?

   (b) What are some useful heuristics (or features) to help resolve anaphora?

Programming

1. NLTK doesn't have much dependency parsing support — there is a little MaltParser interface (`http://maltparser.org/`), but it can be unreliable.

   One popular dependency parsing platform is the Stanford Parser (`https://nlp.stanford.edu/software/lex-parser.shtml`) — the entire package is a somewhat large and requires Java. There are some Python bindings, however. (For example, `http://projects.csail.mit.edu/spatial/Stanford_Parser`)

2. Parse the (tokenised) sentences in `nltk.corpus.treebank_raw.sents()`

3. What proportion of the resulting dependency trees are non–projective? Why do you suppose this is?

Catch-up

- What is a **head**? A **dependency**? How do dependencies differ from **constituents**?
- What are the major differences between the (tree) structure produced by parsing using a context–free grammar, and that produced by parsing using a dependency grammar?
- What are some common dependency arc labels, and what grammatical notions do they encode?